ELSEVIER

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 48 (2012) 702 – 711

Procedia
Social and Behavioral Sciences

Transport Research Arena– Europe 2012

Cooperative Machine-Learning Based Advanced Driver Assistance System for Green Cars

Michalis Masikos[a], Konstantinos Demestichas[a,*], Evgenia Adamopoulou[a], Efstathios Sykas[a]

[a]*Institute of Communication and Computer Systems, 9 Iroon Polytechneiou St., Zografou, Athens GR15773, Greece*

Abstract

It is advocated that the success and user acceptability of Fully Electric Vehicles (FEVs) will predominantly depend on their electrical energy consumption rate and the corresponding degree of autonomy that they offer. FEVs must provide their drivers with the highest possible autonomy, as well as with a high degree of reliability and robustness in terms of energy performance. Thus, appropriate innovative ICT solutions must be adopted, in order to assist the driver in dealing with such energy-related issues, and strengthen FEVs' autonomy and reliability. Such an ICT solution tailored for FEVs is the focus of this paper. In detail, this paper presents a novel implementation of energy-efficient routing based on machine learning engines. It identifies and explains appropriate instance and target attributes, related to road segments and vehicles characteristics. Consequently, it proposes a robust machine learning model, capable of predicting the actual energy that is expected to be consumed by the vehicle on the particular road segment. Scalability aspects are, then, discussed and presented. Finally, useful conclusions on the performance of the proposed model are reached.

Keywords: Fully Electric Vehicles; route calculation; energy efficiency; machine learing; neural networks;

1. Introduction

It is widely accepted that the user acceptability and thus the penetration of Fully Electric Vehicles (FEVs) will predominantly depend on their electrical energy consumption rate and the corresponding degree of autonomy that they offer. FEVs must provide their drivers with the highest possible autonomy, as well as with a high degree of reliability and robustness in terms of energy performance. To that end, appropriate innovative ICT solutions must be adopted, in order to assist the driver in dealing with such

* Corresponding author. Tel.: +30-210-772-1478; fax: +30-210-772-2530
E-mail address: cdemest@cn.ntua.gr

doi:10.1016/j.sbspro.2012.06.1048

energy-related issues, and strengthen FEVs' autonomy and reliability. Such an ICT solution tailored for is described in this article. The goal is to provide an efficient ICT-based solution in this field, by designing and developing a FEV-oriented highly-innovative Advanced Driver Assistance System (ADAS), equipped with suitable monitoring, learning, reasoning and management capabilities that will help increase the FEV's autonomy (distance that can be travelled before battery depletion) and overall electrical energy efficiency.

Several navigation algorithms have been developed in order to inform the vehicle's driver on the shortest path from the current location to a desired destination (Zeng & Yang (2009)). These algorithms usually calculate the route with the shortest distance based on road segment lengths. More advanced algorithms exploit either static travel times (Kaparias & Bell (2009)) (measured by the map provider at some time in the past) or even real-time traffic information (Kaparias et al (2007), Salehinejad & Talebi (2008)), apart from road segments lengths, in order to calculate the route with the shortest-travel time. All of these algorithms are based on the use of a shortest path calculation algorithm (such as Dijkstra's shortest path algorithm (Dijkstra (1959))), which assigns distance-based costs or traffic-based costs on road segments and computes the most cost-effective path. However, so far, none of these algorithms calculates the most energy-efficient route based on actual measurements or actual historical data or employs machine-learning techniques so as to learn from experience.

The calculation of an average consumption on every road segment (in order to adapt the costs of Dijkstra's algorithm) would not provide reliable and accurate results, and would thus not be an ideal solution. The reason for this is that energy consumption depends on numerous parameters, including vehicle's characteristics, weather characteristics, traffic conditions, road characteristics, etc. The construction of a deterministic model that describes this process is a task of extremely great complexity. However, it is still possible to detect certain patterns or regularities through machine-learning. Such patterns can assist in understanding the process, or can be used those to make predictions: Assuming that the future, at least the near future, will not differ significantly from the past moment of sample data collection, the predictions are reasonably expected to be accurate. Moreover, a machine-learning system has also the ability to learn and adapt to changes in order to be always up-to-date.

Prior of the application of machine-learning techniques to a problem, the later must be formally defined. In other words, an appropriate learning model must be constructed. A learning model should be rich enough to capture important aspects of real learning problems, but simple enough to study the problem mathematically. As with any mathematical model, simplifying assumptions are unavoidable. In the present case, the learning model learns how to generate the time spent and the energy consumed on a route based on the existing conditions at each time.

The remaining of the paper is structured as follows: Section 2 describes the machine-learning solution for the energy-efficient routing problem. Specifically, the instance and contextual, as well as the target attributes are identified, while two different approaches of the solution are described. In Section 3 a performance and scalability analysis is conducted in order to demonstrate the feasibility of the proposed system. Finally, Section 5 summarizes the main issued discussed and concludes the paper.

2. The Machine Learning Solution For The Energy-Efficient Routing Problem

Energy efficient routing is one of the pertinent issues related to the autonomy of a FEV. It is based on the realization of dependable energy consumption predictions for the various road segments constituting an actual or potential vehicle route, and it is mainly performed by means of machine-learning functionality, through the use of the so-called Machine-Learning Engines (MLEs). The operation of these engines includes the learning or training process and the scoring or decision process. During the training phase the engine is provided with a set of historical data so as to "learn" from them how to produce reliable estimations in future, yet unseen, situations. Subsequently, the engine is ready to make such predictions, according to current context and based on what it has learned.

The role of the machine-learning system is to provide the user with an energy-efficient optimized route, and in particular to achieve reliable energy cost predictions. In the context of the present work, the on-board routing process is indirectly rather than directly based on machine-learning. This means that machine-learning is applied so as to produce reliable estimations ("predictions") of the energy cost of each road segment. More specifically, the entire routing process is envisioned as follows: (i) The on-board ADAS retrieves a set of alternative routes from the source location to a desired destination. (ii) The total energy cost of each of the alternative routes is determined. This is performed by summing up the energy costs of the road segments consisting an alternative route. The energy cost of each particular road segment is predicted by means of machine-learning. (iii) The most energy-efficient route is selected and presented to the user.

The adoption of the aforementioned approach instead of applying machine-learning in order to predict the energy cost at the whole route is imposed for several reasons.

In case the machine-learning algorithm is applied per route, then the Machine-Learning Engine (MLE) must be provided with the route's segments as input, prior to making a prediction (since a route is defined by its contiguous road segments). Consequently, the MLE must have inputs corresponding to the road segments of the road network of the area of interest. So, as a road network may comprise thousands of links (segments), the following problems arise:

- It is difficult to numerically encode these thousands of different links in a meaningful way.
- Even if this is achieved, an enormous amount of training data (a higher order of magnitude compared to the total number of road segments) is, then, needed. The training (and each re-training) process will be extremely time-consuming.
- Even if training of such a network is accomplished, its predictions are likely to be insufficient. This is because every result will be affected not only by the experience at the same road segment, but also by the experience on different road segments.

In order to reach estimation (prediction) about the energy cost of a road segment, there are two main approaches that can be used. These are summarized in Fig. 1, while the terminology employed is presented in Table 1.

Table 1. Terminology

Term	Explanation
Vehicle Specific Data	vehicle mass, engine type, battery type, etc.
Context Data	type of day (e.g., holiday, working day), time of day (e.g., morning, noon), month (e.g., January), temperature (e.g., hot, cold), humidity (e.g., rainy), etc.
Electrified Aux Data	type and status (on/off) of electrified auxiliaries, such as lights, A/C, radio, etc.
NECF	Normalized Energy Cost Factor, i.e. a cost factor that corresponds to the energy cost of a particular road segment and can be used directly by heterogeneous vehicles
PNECF	Predicted Normalized Energy Cost Factor
AEC	Actual Energy Cost, i.e. actual energy expenditure on a specific road segment, e.g. measured in kWh
PAEC	Predicted Actual Energy Cost
MLE	Machine-Learning Engine (such as neural network, or other kind of Machine Learning technology)

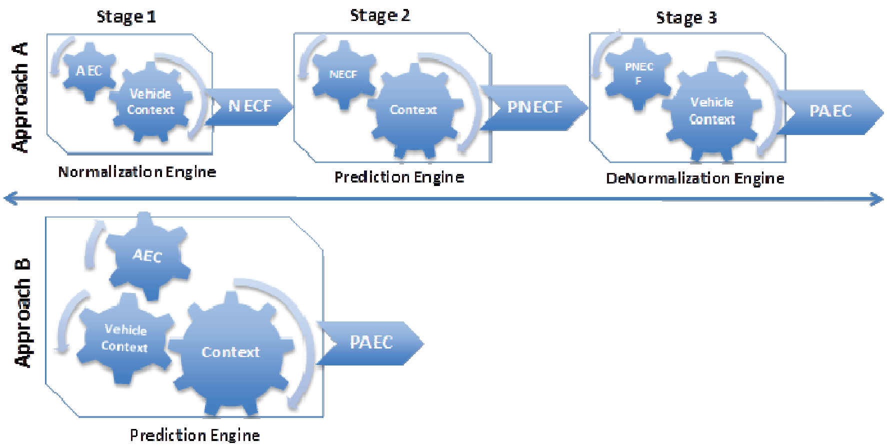

Fig. 1. Flow Charts for the Energy Prediction Approaches

2.1. Approach A

A MLE is used to transform the energy consumed by a vehicle on a road segment into a neutral energy cost factor. This neutral factor can be shared among vehicles without any arising problem. Several such factors (corresponding to the same road segment) can be used to train a MLE, which - once trained - can be used to produce reliable predictions about the road segment's normalized energy cost factor. Whenever such a prediction is made by this MLE, a "reverse" process must subsequently follow, transforming the predicted normalized energy cost factor into an actual energy value.

2.1.1. Stage 1

It is assumed that the vehicle has travelled on a particular road segment and has measured the AEC on this segment recording this AEC value, together with the associated Context. A MLE is employed, taking as input the AEC together with vehicle specific data and producing a NECF for this road segment. Hence:

{Vehicle Specific Data, AEC} → *NECF*

The NECF is related to the same context as the corresponding AEC value. Thus, the following data pieces can now be shared by the vehicle, i.e. sent to other vehicles via V2V or via V2I/I2V:

{Context Data, NECF}

These data are normalized, facilitating their use as training data by other vehicles. Details on this process are described in the following paragraph. It should be, also, noted, that by using the neutral NECF values no vehicle specific data are needed to be shared.

2.1.2. Stage 2

Let us assume that another vehicle has gathered such data originating from its own measurements as well as from various other vehicles, i.e. data of the following form:

$Context_1$ $NECF_1$
$Context_2$ $NECF_2$
…
$Context_n$ $NECF_n$

It should be kept in mind that all of the above data refer to the same road segment. These data can be used for training a MLE. After training, this MLE will be in position to predict the NECF of the particular road segment whenever this is required. Hence, after it has been trained, this MLE can be employed to produce a reliable PNECF for the current Context, i.e.:

{Context} → PNECF

Note that a different (trained) MLE is used for each particular road segment.

2.1.3. Stage 3

This predicted normalized energy cost value can now be used as input by a different MLE, together with vehicle's specific data, in order to infer the actual energy that will be consumed on the road segment under consideration. Hence:

{PNECF, Vehicle Specific Data} → PAEC

The resulting PAEC value represents the actual energy that is expected to be consumed by the vehicle on the particular road segment.

2.2. Approach B

This approach is more straightforward, but requires the sharing of vehicle specific data among the enabled vehicles. This approach does not make use of normalized energy cost factors.

In particular, in this approach, a MLE is trained based on the following form of training data:

Vehicle specific data$_1$ *Context*$_1$ *AEC*$_1$
Vehicle specific data$_2$ *Context*$_2$ *AEC*$_2$
...
Vehicle specific data$_n$ *Context*$_n$ *AEC*$_n$

Then, this (trained) MLE is employed in order to predict the actual energy cost of a road segment, taking into account not only the current Context, but the vehicle specific data as well:

{Context, Vehicle Specific Data} → PAEC

Again, a different MLE is needed for each road segment.

For the system under consideration, both of the aforementioned approaches are sufficient, however the first one (Approach A) also presents the noteworthy advantage that vehicle specific data need not be transmitted, and from this perspective seems more suitable.

In the context of the present work, an excerpt of a probable set of training data is depicted in Fig. 2. Fig. 3 concisely illustrates how each type of the data included in the table is used.

Each row of data refers to a specific road segment, which can be referenced by a source point and an end point. Hence, a road segment Ai→ Aj has point Ai as the starting point, and point Aj as the end point. Of course, there can be multiple rows of data referring to the same road segment, as depicted in the table. Each row comprises data that have been measured, and more specifically: a set of instance attributes (such as the time band or the type of day when the measurement was collected) and a set of target of attributes (such as the energy and time spent on the road segment). When enough data concerning a road segment have been gathered, these are used to train a machine-learning engine (e.g., a specific type of neural network).

Once the training is complete, the machine-learning engine can then be used to predict (estimate) the value of a target attribute that is, of course, unknown a priori (e.g. energy that is expected to be spent on a particular road segment); obviously, only the values of the instance attributes are known when attempting to make such a prediction (estimation). Fig. 2 depicts such a case, in which a prediction is required; the instance attributes are known and are fed to the machine-learning engine as input, and (one or more of) the target attributes are unknown and their values are expected to be produced by the machine-learning engine as output.

Regarding the target attributes depicted in Fig. 2, it is valuable to note that the construction of a MLE that predicts all these attributes simultaneously is better to be avoided. This is because such a MLE would not predict the most energy efficient segment, but the most optimal segment regarding time cost, energy cost and average speed simultaneously. The wisest choice is to study each target attribute separately, although studying the interactions of these attributes might prove useful as well.

In Table 2, a sample set of all the instance attributes that are going to be examined is presented.

Contextual level attributes		Predictive level attributes													
		Instance attributes									Target attributes				
		Time context			Dist.	Vehicle context			Weather context						
Start point	End point	Time band	Day type	Month	Dist.	Vehicle mass	Elec. aux.1	Elec. aux.2	Temp.	Hum.	Average speed	Time cost Actual	Indicator	Energy cost Actual	Indicator
A_i	A_j	Morning	Mon	Jan	2 km	1.0-1.2 ton	On	Off	Hot	Low	51 km/h	2.5 min	0.7	3.1 KWh	0.6
A_i	A_j	Morning	Mon	Jan	2 km	1.0-1.2 ton	On	Off	Hot	Low	49 km/h	2.6 min	0.72	3.2 KWh	0.61
A_i	A_j	Afternoon	Sat	Feb	2 km	1.0-1.2 ton	Off	Off	Cold	High	54 km/h	2.2 min	0.69	2.9 KWh	0.58
…															
A_i’	A_j’	Morning	Mon	Jan	2.3 km	1.0-1.2 ton	On	Off	Normal	Low	66 km/h	2.5 min	0.7	3.15 KWh	0.61
A_i’	A_j’	Morning	Mon	Feb	2.3 km	1.0-1.2 ton	On	Off	Hot	Low	66 km/h	2.6 min	0.72	3.25 KWh	0.63
A_i’	A_j’	Afternoon	Holiday	Feb	2.3 km	1.0-1.2 ton	Off	Off	V. Cold	V. High	68 km/h	2.2 min	0.69	2.98 KWh	0.59
…															
Prediction Example															
A_i	A_j	Morning	Monday	Feb	2 km	1.0-1.2 ton	On	Off	Hot	Low	?	?	?	?	?
…															

Fig. 2. Example of Road Segments Attributes

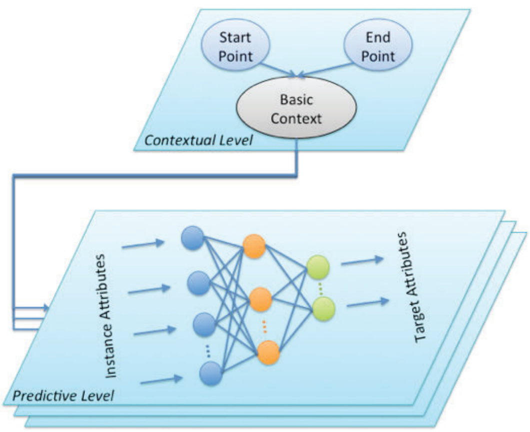

Fig. 3. A generic machine-learning model and the basic attributes that it comprises

Table 2. Instance Attributes

Instance and contextual-level attributes		Data description	Source
Time Context	Time band	Time band is significant, as it resembles people habits. For example, all people go to work or leave from there almost at the same time.	Internal clock (together with Internet service for synchronization): time context can be retrieved from an internal clock of the ADAS, and one of the publicly available Internet Time Servers can be used for clock synchronization.
	Day type	The same is valid also for day type. For example, people usually do not work at weekends.	
	Month	The same is valid also for month. For example, the summer months usually correspond to people vacations.	
Vehicle Context	Weight	The vehicle weight affects the consumption.	CAN bus or manual configuration: these attributes can be retrieved either from the manufacturer's site or from the vehicle's electronic units, or alternatively they can be manually configured.
	Avg Cons.	This attribute refers to vehicle's consumption rates.	
	Brand	It reflects manufacturers' different technologies.	
	Model	Model is important as it reflects the vehicle's dimensions.	
	Battery Health	Battery health may be defined in a % percentage and it affects the consumption function.	CAN bus: these attributes can be retrieved from the vehicle electronic units.
	Battery Capacity	This attribute is also important as bigger batteries have usually better consumption behaviour.	
	SOC	This attribute is important as battery performance is not linear and depends on SOC.	
	Electric Aux (n) Status	The use of any electric auxiliary (n) affects energy consumption.	
Weather Context	Temperature	The temperature affects the energy consumption directly by altering battery's behaviour and indirectly by using auxiliary electric systems.	Internet or CAN bus: these attributes can be retrieved either from Internet weather sites or from vehicle sensors.
	Humidity	Increased humidity means either rain or fog, which change driving behaviour and demand the operation of auxiliary electric systems (e.g. fog lights or windscreen wipers).	
Road Segment Attributes	Segment ID	This attribute defines the road segment unique identifier.	ADAS map static info: these attributes are retrieved from the ADAS map with the use of map position services provided by the underlying SDK.
	Segment Start Point	This attribute defines the road segment start point.	
	Segment End Point	This attribute defines the road segment end point .	
	Distance	This attribute may be also retrieved from vehicle's odometer.	
	Inclination (slope)	This attribute affects the energy consumption positively (downwards) or negatively (upwards).	
	Capacity	This attribute refers to the width of the road.	
	Type	This attribute refers to the kind of the road containing the segment, e.g. national road, highway, city road etc.	

3. Performance and Feasibility Analysis

In the following, we conduct an analysis, from a performance scope of view, on the feasibility and scalability of the use of Machine-Learning Engines for energy efficient routing. In the context of the present paper, MLE implementation is based on Artificial Neural Networks. Two different network structures of multilayer perceptron (MP) are tested. The first network has one hidden layer consisting of 5 neurons, and the second one has two hidden layers with 10 and 2 neurons respectively. In both cases, the number of input nodes at the input layer has been set to be the same and equal to 6 nodes. Input parameters reflect the context in which the vehicle is operating, such as time of day, month, etc. Input parameters can be selected from the list of Table 2. As it has been mentioned, the outputs are the predicted energy consumption and the predicted travel time (2 nodes). Hence, the tested ANN structures can be denoted as 6-5-2 and 6-10-2-2. Test networks with more than 2 hidden layers are not tested, as it is generally accepted that commonly they cause extra training delay but no actual improvement in predictions (Chiang et al (2006), Jain & Nag (2007)). Unless it is mentioned otherwise, the default ANN structure that is being in use is the first one, i.e. 6-5-2.

Moreover, the logistic activation function for the hidden layers and the linear activation function for the output layer is used, complying to a typical ANN setup. The training method used is the scaled conjugate gradient (Moller (1993)). During the training process, the model is tested and validated with a random 20% subset of the training data.

The diagram in Fig. 4(a) displays the training time of the MP "6-5-2" vs. the size of the training dataset. The training procedure takes place for 10 different sizes of training datasets, ranging from 100 to 1000 measurements. As observed, in general, the training time rises as the size of the training dataset increases. However, this observation is not entirely valid for all types of datasets, specifically for those containing 700, 900 and 1000 measurements. This is because the quality of the data that are included in these three dataset types can happen to be better, and for that reason the training time can be shorter than expected. As can generally be deduced from the diagram, the training time in all of the cases is less than 2 seconds for datasets with up to one thousand measurements and such a delay is acceptable.

Another important study is depicted in the diagram in Fig. 4(b). It displays the training time of two neural networks with different internal structures. As can be observed for small training datasets the difference in training time is negligible. The situation alters for larger datasets (larger than 500 measurements). The time delay is almost double for the "6-10-2-2" network compared to the "6-5-2" network in case of 1000 measurements. Based on the test results, it can be concluded, that either of the two ANNs for training datasets up to 500 measurements can be deployed.. If it is desirable to use a greater training dataset, then it is advisable to use ANNs with simpler structures, such as the "6-5-2" MP.

The ADAS has to propose to the driver the most energy efficient route. For that reason, it has to evaluate the energy efficiency (predicted energy consumption) of several candidate road segments, so as to detect those that minimize the total energy expenditure to get to the desired destination. Within this context, in the next diagram, it is assumed that, in order to find the most energy efficient route, a set of road segments (in particular, the MLEs pertaining to these road segments) has to be initially trained. Sets of three different sizes, a small-sized (containing 100 road segments), a medium-sized (containing 300 road segments), and a larger one (containing 500 road segments) are considered. As can be observed in Fig. 4(c) the total training delay is restrained under 2 minutes for cases of relatively low dataset sizes (100 measurements), or relatively low number of road segments. For cases other than these, the total training delay can get significantly higher. This means that the system would greatly benefit in its performance from background (or offline) training. By background training it is suggested to enable the system to train its MLEs in the background, at times prior to their actual use for scoring (as opposed to performing the training at the time of the user request, a solution that can be denoted as online training). This shall lead to a great performance enhancement and to a reduction in the time the user has to wait until the optimal route is presented to him. By enabling background training, it can be safely assumed that –at the time of the user request– the number of untrained MLEs would be confined to a relatively low number, e.g. 100

road segments or below, even for large routes. In this way, the need for online training is reduced and the corresponding computational load can be handled within reasonable time constraints.

Fig. 4. Training Times (a) MP "6-5-2"; (b) MP "6-5-2" vs. MP "6-10-2-2"; (c) MP "6-5-2"; (d) Medium-Sized Route

Finally, a medium-sized set of road segments is considered and the total training time of the "6-5-2" MPs and the "6-10-2-2" MPs for the entire set is compared (Fig. 4(d)). Again, it is observed that the training delay almost doubles for the higher complexity neural network when using training datasets larger than 500 measurements, but remains comparatively the same for lower dataset sizes. Hence, it is advisable to combine large datasets with MPs with one hidden layer. As previously, the system would greatly benefit in its performance from background training.

The diagram of Fig. 5 reports the total scoring time for predicting the energy consumption and the travel time of the three sets of road segments, using MPs with different network structures. The tested network structures are the same as in the training process. The scoring time is very close to zero, even for the more complex network, and therefore the network complexity has insignificant affect on the scoring time. Based on the same diagram, it can be deduced that no factor has significant negative impact on the ANN scoring time, and that the total time required for scoring is negligible.

4. Conclusions

At the heart of the proposed energy-driven routing algorithm lies the machine-learning functionality. Machine-learning generally involves two separate processes, training (or learning) and scoring (or decision). Section 3 investigated the performance and scalability of both processes. Scalability studies were performed by deploying appropriate Artificial Neural Networks. Various parameters were also considered in the studies, including different training dataset sizes, MLE complexities, etc.

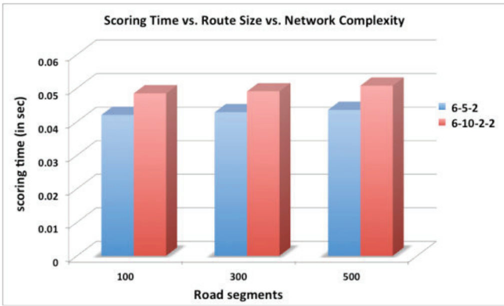

Fig. 5. Scoring Time

The primary conclusions that were reached are the following: i) from the two processes, training and scoring, the first is by far the most computationally challenging; ii) it was found that ANNs achieve a high degree of scalability; iii) training can be achieved in less than 3 minutes for a set of 100 road segments, in a wide range of cases; iv) background training can help the system be instantly ready for when scoring is required; v) larger training datasets are best to be combined with less complex ANNs; vi) generally, using a larger dataset for the training process results in higher training times, which implies that it is better to use larger training datasets only when the quality of predictions achieved by the MLE is expected to rise significantly, and vii) scoring is a perfectly scalable process in the ADAS. According to the presented approach, an ADAS will learn from actual driving experience on different routes. This approach can be applicable to both FEVs and non-FEVs (Internal Combustion Engine vehicles), although the authors believe that the application into FEVs consists a more exciting challenge.

Acknowledgements
This work has been performed under project EcoGem, which has received research funding from the European Community's Seventh Framework Programme. This paper reflects only the authors' views, and the Community is not liable for any use that may be made of the information contained therein.

References

Chiang, W., Zhang, D., & Zhou, L., (2006). Predicting and explaining patronage behavior toward web and traditional stores using neural networks: a comparative analysis with logistic regression. *Journal of Decision Support Systems*, 41, 514-531.

Dijkstra, E., W., (1959). A note on two problems in connexion with graphs. Journal of *NUMERISCHE MATHEMATIK*, 1, 269-271.

Jain, B., & Nag, B., (2007). Performance evaluation of neural network decision models. *Journal of Management Information Systems*, 14, 201– 216.

Kaparias, I., & Bell, M.G.H., (2009). Testing a reliable in-vehicle navigation algorithm in the field. *IET Journal of Intelligent Transport Systems*, 3, 314-324.

Kaparias, I., Bell, M.G.H., Chen, Y., & Bogenberger, K., (2007). ICNavS: a tool for reliable dynamic route guidance. *IET Journal of Intelligent Transport Systems,* 1, 225 – 233.

Li, Q., Zeng, Z, & Yang, B., (2009). Hierarchical model of road network for route planning in vehicle navigation systems. *IEEE Intelligent Transportation Systems Magazine*, 1, 20 – 24.

Moller, M., F., (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Journal of Neural Networks*, 6, 525--533.

Salehinejad, H., & Talebi, S., (2008). A new ant algorithm based vehicle navigation system: A wireless networking approach. *International Symposium on Telecommunications (IST 2008)*, 36-41.